

49986-0504 (RSID 1-330)

Patent

UNITED STATES PATENT APPLICATION

FOR

MULTIPLE PAGE-RANGE PRINTER SETTINGS

INVENTOR:

ZHONGMING YU

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CA 95125-5106
(408) 414-1080

EXPRESS MAIL CERTIFICATE OF MAILING

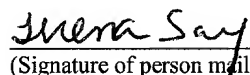
"Express Mail" mailing label number EL734778779US

Date of Deposit March 30, 2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Box Patent Application, Commissioner for Patents, Washington, D.C. 20231.

Tirena Say

(Typed or printed name of person mailing paper or fee)



(Signature of person mailing paper or fee)

MULTIPLE PAGE-RANGE PRINTER SETTINGS

COPYRIGHT AUTHORIZATION

A portion of the disclosure of this patent document contains material which
5 is subject to copyright protection. The copyright owner has no objection to the
facsimile reproduction by any one of the patent disclosures, as it appears in the U.S.
Patent & Trademark Office patent files or records, but otherwise reserves all
copyright rights whatsoever.

10 FIELD OF THE INVENTION

The present invention generally relates to the printing of electronic
documents. The invention relates more specifically to a mechanism for printing
multiple page-ranges within an electronic documents.

15 BACKGROUND OF THE INVENTION

Computers and output devices such as printers are extensively used in word
processing and other document management tasks. In general, each computer is
typically connected to one or more printing devices. The connection may be over a
dedicated line that is established directly between the computer and the printing
20 device or instead, through a shared network, such as a company's internal network or
LAN that allows the printing device to be shared by multiple computers connected to
the network. In general, to allow a computer to communicate with a particular
printing device, a print driver that is associated with the specific printing device is
installed on the client computer. Each print driver includes a set of commands that
25 allow applications executing on the client to communicate with the corresponding
printing device.

Conventionally, when a user selects a print option to print an electronic document, the user is presented with an interface window that allows the user to select or enter certain parameters for printing the electronic document. For example, FIG. 1A illustrates a conventional print interface window 100 that is displayed to a user in response to the user selecting a print option. As depicted, print interface window 100 includes, among other things, a page range selector 104 and a properties button 102. The page range selector 104 allows a user to select a range of pages that are to be sent to a printing device for generating a printed copy (hard-copy) of the selected pages. For example, using page entry box 106, the user may select pages 1, 3 and 5-12 for printing by the printing device that is identified in printer name entry box 108.

Additionally, in response to selecting the properties button 102, the user is presented with a properties window 150 as depicted in FIG. 1B. By interfacing with the properties window 150, the user may select the paper size format (for example, Letter 8½ x 11 in., Legal 8½ x 14 in., A4 210 x 297 mm), and the paper source (i.e., Paper Tray) that is to be used for printing the electronic document.

For example, using paper size drop-down window 158, the user can choose from a set of preselected paper formats. Once selected, the paper format is then used to print all pages within the range of pages selected in page entry box 106.

Also, by selecting “Tray2” in entry box 156, the user may cause the selected pages to be printed using whatever media type that happens to be currently loaded in Tray 2 of the printing device. In addition, by selecting the “Different Paper for 1st Page” option 152, the user may select the paper source that is to be used for printing the first page of the selected range of pages. For example, by selecting “Tray1” in entry box 154, the user may cause the first page of the selected range of pages to be printed using the media type that is currently loaded in Tray 1 of the printing device.

However, a significant drawback with using a conventional printing mechanism as depicted in FIGs. 1A and 1B is that the user can only select a single paper size format, thus requiring that all pages within the selected page range be printed using the same paper size format.

5 In addition, a further drawback with using a conventional printing mechanism is that the user is limited to only two paper sources for printing the pages of an electronic document, and in the case of pages 2-n of the selected page range (“base pages”), the user is limited to only a single paper source. Furthermore, the user can only select the specific tray that is to be used as a paper source for printing
10 the document. Thus, the user is not provided with any information as to the type of media that is associated with a specific tray selection (for example, 8 ½ x 11 paper, 8 ½ x 14 paper, recycled paper, transparencies, color paper, etc.), let alone the actual type of media that is available for generating a hard copy of the electronic document.

In addition, because the conventional printing mechanism fails to provide the
15 user with any information as to the media type that is currently loaded with a particular paper source, the user can mistakenly select a paper size format that is inconsistent with the media type that is currently loaded with a particular paper source.

Based on the foregoing, there is a clear need for a printing mechanism that
20 allows a user to flexibly select the media types that are used to print a hard copy of a particular range of pages within an electronic document.

There is also a need for a printing mechanism that provides a user with dynamic information as to the media types that are currently available to the user for printing a hard copy of an electronic document.

25 There is also a need for a printing mechanism that allows a user to select multiple paper size formats for printing pages within a selected page range.

There is also a need for a printing mechanism that identifies to the user the set of one or more paper size formats that are compatible with the media types that are currently associated with the printing device.

- 5 Still further, there is a clear need for a printing mechanism that will allow a user to select multiple paper sources for printing the pages within a multiple base page range of an electronic document.

SUMMARY OF THE INVENTION

A method and system for printing an electronic document is disclosed. In one aspect, a first media type for printing a first page range of one or more pages of an electronic document is selected. A second media type for printing a second page range of one or more pages of the electronic document is also selected. Information that identifies the first and second media types for printing the first and second page ranges of one or more pages of the electronic document are transmitted to a printing device.

According to one feature, in response to receiving the information, the printing device generates a printed copy of the electronic document that includes the first and second page ranges; the first page range being printed using the first media type and the second page range being printed using the second media type.

According to another aspect, a method and system for printing an electronic document is disclosed whereby a request is received for delivery of a set of print attributes that are available for printing said electronic document on said printing device. User interface data, that identifies the set of print attributes that are available for printing said electronic document on said printing device, is generated. The user interface data is transmitted to a client device for displaying the set of print attributes that are available for printing said electronic document on said printing device.

In other aspects, the invention encompasses a computer apparatus, a computer-readable medium, and a carrier wave configured to carry out the foregoing steps.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

5 FIG. 1A is a conventional print interface window that is displayed to a user in response to the user selecting a print option;

 FIG. 1B is another conventional print interface window that is optionally displayed to a user;

 FIG. 2 illustrates a block diagram of a document printing system in which the
10 invention may be utilized;

 FIG. 3 is an example of a multiple page-range printing interface window that may be displayed to a user;

 FIG. 4 is another example of a multiple page-range printing interface window that may be displayed to a user;

15 FIG. 5 is another example of a multiple page-range printing interface window that may be displayed to a user;

 FIG. 6 is another example of a multiple page-range printing interface window that may be displayed to a user;

20 FIG. 7 is another example of a multiple page-range printing interface window that may be displayed to a user;

 FIG. 8 is another example of a multiple page-range printing interface window that may be displayed to a user;

 FIG. 9A is a table that illustrates an example of the type of information that may be collected by the printing mechanism based on FIGs. 3-8;

FIG. 9B is another table that illustrates a further example of the type of information that may be collected by the printing mechanism based on a different set of selections than those described in FIGs. 3-8;

FIG. 10 is a flow diagram that illustrates an example of a sequence that can
5 be used for processing multiple page-range print requests within an electronic document; and

FIG. 11 illustrates an example of a computer system that may be used to provide the described printing mechanism.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A method and system for printing an electronic document is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It
5 will be apparent, however, to one skilled in the art that the invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the invention.

10

OPERATIONAL CONTEXT

A printing mechanism is provided for generating a hard copy of pages that are contained in an electronic document. In one embodiment, when a user selects a print option to print an electronic document, the user is presented with an interface window that allows the user to select multiple page-ranges within the electronic
15 document. The user may also select from a variety of print options that may include specific print characteristics, for example simplex/duplex modes or the ink color, that are to be applied to one or more of the multiple page-ranges; specific media types that are to be used to print the pages of one or more of the multiple page-ranges; or specific paper sources that are to be used to print the pages of one or more of the
20 multiple page-ranges.

In one embodiment, in response to a user selecting a print option at a client, a print message is sent from the client to a printing device to request information about the specific media types that are currently available for printing pages of the electronic document. In response to receiving the print message, the printing device
25 returns current media type information to the client that identifies the media types that are currently available at the printing device. Based on the current media type

information, an interface window is generated and presented to the user at the client. The user may then select specific media types that are to be used to generate a hard copy of selected page-ranges within the electronic document.

5 In certain embodiments, upon receiving a print message, the printing device generates a print window interface based on the specific characteristics of the selected printing device. The selected printing device then forwards the print window interface to the client computer for display to the user.

FIG. 2 illustrates a block diagram of a document printing system 200 in which the invention may be utilized. The system 200 generally includes a printing device 202, one or more computing devices (Clients 204,220), and a network 206.

10 The computing devices 204 and 220 each comprise a workstation, personal computer, lap-top computer, hand-held device, often referred to as a "Highly Constrained" device (e.g. cellular phones, Palm Pilots, PDAs, etc.) or other similar type of devices that can be configured to communicate with printing device 202. In this example, client 204 includes a document application 216 and a browser process 218. The document application 216 provides a mechanism for generating and viewing electronic documents. For example, document application 216 may represent any number of software applications which may include but is not limited to Microsoft Word®, WordPerfect®, Microsoft Excel®, Power Point, Lotus Notes, Adobe Acrobat, etc. Thus, computing devices 204 and 220 may contain or have access to, an assortment of different electronic document information that has been created in a variety of different file formats.

25 Browser process 118 represents a browser type application, such as Netscape Navigator®, Microsoft Internet Explorer®, or other similar type of WAP or HTML browser application that may be used to communicate with printing device 202 in a client-server relationship. In one embodiment, browser process 218 provides a

display interface that allows a user to select from a variety of different print options for printing pages of an electronic document on printing device 202. In one embodiment, browser process 118 provides a user interface for selecting multiple page-ranges of an electronic document and the specific media types that are to be
5 used by printing device 202 to print the pages within each page-range.

Network 206 provides a communication link between printing device 202 and the computing devices 204 and 220. Network 206 may represent any number of communication mechanisms which may include but is not limited to the global packet-switched network known as the Internet, a private Intranet for a particular
10 company, a dedicated serial or parallel link, a wireless communication link, a telephone or cable line, or any combination thereof. The network 206 may form part of a LAN or WAN and may use a variety of different communication protocols.

The printing device 202 is configured to communicate with clients 204 and 220 over network 206 and to collect or generate print characteristic data based on the
15 specific characteristics of the printing device 202. In this example, printing device 202 is a multifunctional printer or multifunctional peripheral (MFP) that includes, among other things, a media type ID process 208, a print process 210 and a server process 214.

Media type ID process 208 provides a mechanism for entering the type of
20 media that is currently loaded in each tray of printing device 202. In one embodiment, Media type ID process 208 generates a set of media characteristic data that contains specific information as to the media type that is available for printing documents on printing device 202. In certain embodiments, Media type ID process 208 is coupled to an operational panel that provides an interface which allows a user
25 to input information about specific characteristics of printing device 202. For example, printing device 202 may include an operational panel that can be used to

enter information about the media types that are loaded into the different trays of printing device 202.

Print process 210 provides a mechanism for printing selected pages of electronic documents that are received from clients 104 and 120 over network 106.

5 In one embodiment, print process 110 is configured to receive information identifying a particular media type that is to be used to print a range of pages within an electronic document. Based on the identified media type, print process 110 selects the appropriate paper source and uses the media within the paper source to generate a hard-copy printout of the pages within the range of pages.

10 Server process 114 is configured to receive print requests from clients 104 and 120. In certain embodiments, in response to receiving a print request, server process 114 generates interface data for displaying a multiple page-range interface window on the requesting client. In one embodiment, the interface data includes information about the specific characteristics of the printing device, for example the
15 currently available media types which are generally not available in a conventional printing interface. By interacting with the multiple page-range interface window the user may select specific media types, specific paper sources and/or specific print characteristics that are to be applied to multiple page-ranges within an electronic document.

20 In certain embodiments, server process 114 is configured as a Web server that provides a communication mechanism for communicating with clients via network 206. In one embodiment, the Web server includes a Hypertext Transfer Protocol (HTTP) daemon that can respond to requests from clients 104 and 120 and establish an HTTP connection between the printing device 202 and the requesting
25 client. HTTP is merely an example of a communications protocol that can be used

in an embodiment. Other protocol that facilitates exchange of information among a client and server can be used.

MULTIPLE PAGE-RANGE PRINTING INTERFACE

5 A multiple page-range printing interface is provided that allows a user to select multiple page-ranges within a document and to choose individual print attributes for printing the pages within each page range. In one embodiment, information defining the multiple page-range printing interface is provided as an HTML or XML document to either client 204 or 220 from printing device 202. In
10 certain embodiments, the multiple page-range printing interface includes one or more print attribute options that are based on specific characteristics of the selected printing device. For example, the multiple page-range printing interface may include a list of the different media types that are currently available to the user based on the particular characteristics of the printing device that has been selected. In one
15 embodiment, a mapping is maintained between the available media types and the one or more paper sources that contain each of the available media types.

 In certain embodiments, in response to a change in the characteristics of the selected printing device, the multiple page-range printing interface is dynamically updated to reflect the current characteristics of the selected printing device. For
20 example, in response to a new media type being loaded into “tray 1” of the selected printing device, the multiple page-range printing interface may be automatically updated to reflect a new list of media types that are currently available to the user based on the current characteristics of the selected printing device. Additionally, if the user selects a different printing device, the multiple page-range printing interface
25 automatically updates to display the set of print attribute options that are currently available based on the newly selected printing device.

FIG. 3 illustrates an example of a multiple page-range printing interface 300 that may be displayed by browser process 218 on client 204. In this example, interface 300 includes a total page entry box 304, a begin range entry box 306, a set of selected print attribute boxes 324, and a set of print attributes option boxes 326.

- 5 In one embodiment, printer characteristic information that is not generally available to the client, such as the media type that is currently loaded in the printing device, is dynamically provided by printing device 202 for display within multiple page-range printing interface 300.

- Total page entry box 304 allows the user to enter the total number of pages that are to be printed within the electronic document. For example, if the user wants to print a hard copy of the first one-hundred pages of a document, the user enters the value “100” in total page entry box 304. In one embodiment, by entering a value of zero (“0”), the multiple page-range mechanism is disabled, thus causing the settings under the “Multiple Page-range Setting” tab 305 to be ignored by the system. For example, in certain embodiments, a “Page Setup” tab 307 and an “Advanced” tab 309 provide a mechanism whereby the user may select from a set of conventional print options (for example, conventional options previously depicted in FIGs. 1A and 1B), for printing an electronic document. In certain embodiments, by entering a value of zero (“0”) in total page entry box 304, the set of conventional print options are used for printing pages of the electronic document.
- 10
15
20

Alternatively, by entering a non-zero value in total page entry box 304, the user is presented with an interface (for example multiple page-range printing interface 400 in FIG. 4) for entering one or more page-ranges and the set of printing attributes that are to be applied to each range of pages.

- 25 Begin range entry box 306 enables the user to select a specific range of pages for which the selected printing attributes will apply. How a user may select multiple

page-range settings using begin range entry box 306 is described in detail below using the examples in FIGs. 4-11.

5 In this example, the set of print attributes option boxes 326 include a set simplex/duplex option box 314, a set paper source option box 316 and a Set media type option box 318.

10 The set simplex/duplex option box 314 enables a user to print the selected range of pages using either simplex or duplex mode. Simplex/duplex selected box 308 indicates the current selection that the user has made for the selected page range. In one embodiment, by selecting printer default option 302 in set simplex/duplex option box 314, the user can choose to have the selected range of pages to be printed using the default setting of the selected printing device.

15 Set paper source option box 316 enables a user to select a specific paper source that is to be used in printing the selected range of pages. Paper source selected box 310 indicates the paper source that the user has selected for printing the particular range of pages.

20 Set media type option box 318 enables a user to select a specific media type that is to be used in printing the selected range of pages. In one embodiment, set media type option box 318 automatically updates to reflect the media types that are currently loaded in the selected printing device. Media type selected box 312 indicates the media type that the user has selected for printing the particular range of pages.

25 The OK button 320 allows the user to initiate the printing of the one or more selected page-ranges based on the print attributes that have been selected by user. The Cancel button 322 allows the user to cancel the print request, thus causing the selected page-ranges not to be printed.

SELECTING PAGE-RANGES AND THE PRINT ATTRIBUTES THAT APPLY

As previously indicated, by entering a non-zero value in total page entry box 304, the user is presented with an interface for entering one or more page-ranges and the printing attributes that are to be applied to each of the page-ranges. FIG. 4 illustrates an example multiple page-range printing interface 400 that may be displayed by browser process 218 on client 204 in response to a user entering a non-zero value in total page entry box 304. In this example, a user has entered a value of eight ("8") in total page entry box 304, thus indicating that a total of eight ("8") pages are to be printed. As further depicted, the user has selected a value of one ("1") in begin range entry box 306 to indicate that the current page-range is to start with page one ("1") of the electronic document. In one embodiment, the printing mechanism determines that the page-range associated with multiple page-range printing interface 400 currently consists of pages 1-8 (e.g., the value of begin range entry box 306 to the value of total page entry box 304).

Simplex/duplex selected box 308 indicates that the user has selected simplex option 402 in set simplex/duplex option box 314 for the selected page range; paper source selected box 310 indicates that the user has selected "tray 2" option 404 in set paper source option box 316 as the paper source for the selected page range; and media type selected box 312 indicates that the user has selected "75 g/m² (White)" option 406 in set media type option box 318 as the media type for the selected page range.

In one embodiment, in response to selecting a specific media type in set media type option box 318, the paper source selected box 310 is automatically updated to reflect the paper source that contains the selected media type in the printing device. For example, in response to the user selecting light green option

410, if “tray 1” of the printing device represents the paper source that is currently loaded with light green paper, the paper source selected box 310 is automatically updated to indicate a selected paper source of “tray 1”. In certain embodiments, the multiple page-range printing interface may only provide for selecting the specific media type (set media type option box 318 and media type selected box 312) and thus not include a selection option for selecting the specific paper source (i.e., set paper source option box 316 and paper source selected box 310).

After entering the specific attributes that are to be used for the current page-range, the user may select by pressing or clicking the next range button 408 to enter additional page-ranges and the printing attributes that are to be applied to each of the additional page-ranges.

For example, FIG. 5 illustrates an example multiple page-range printing interface 500 that may be displayed by browser process 218 on client 204 in response to user selecting the next range button 408. In this example, the user has selected a value of two (“2”) in begin range entry box 306 to indicate that the current page-range is to start with page two (“2”) of the electronic document. In one embodiment, the printing mechanism uses the new value in begin range entry box 306 in determining the last page of the previous page-ranges that have been entered. For example, by entering the value “2” in begin range entry box 306 of multiple page-range printing interface 500, the printing mechanism determines that the page-range associated with multiple page-range printing interface 400 consists of pages 1-1. Additionally, the printing mechanism determines that the page-range associated with multiple page-range printing interface 500 currently consists of pages 2-8.

Further to this example, the simplex/duplex selected box 308 indicates that the user has selected duplex option 502 in set simplex/duplex option box 314 for the selected page range; the paper source selected box 310 indicates that the user has

selected “Manual Feeder” option 504 in set paper source option box 316 as the paper source for the selected page range; and the media type selected box 312 indicates that the user has selected “Recycle” option 506 in set media type option box 318 as the media type for the selected page range.

5 Again, after entering the specific attributes that are to be used for the current page-range in multiple page-range printing interface 500, the user may select the next range button 408 to enter additional page-ranges and the printing attributes that are to be applied to each of the additional page-ranges.

FIG. 6 illustrates another example of a multiple page-range printing interface 10 600 that may be displayed by browser process 218 on client 204 in response to user selecting the next range button 408 in multiple page-range printing interface 500. In this example, the user has selected a value of five (“5”) in begin range entry box 306 to indicate that the current page-range is to start with page five (“5”) of the electronic document. In one embodiment, the printing mechanism uses the new value in begin 15 range entry box 306 in determining the last page of the previous page-ranges that have been entered. For example, by entering the value “5” in begin range entry box 306 of multiple page-range printing interface 600, the printing mechanism determines that the page-range associated with multiple page-range printing interface 500 consists of pages 2-4. Additionally, the printing mechanism determines that the 20 page-range associated with multiple page-range printing interface 600 currently consists of pages 5-8.

Further illustrated in this example, the simplex/duplex selected box 308 indicates that the user has selected simplex option 602 in set simplex/duplex option box 314 for the selected page range; the paper source selected box 310 indicates that 25 the user has selected “Tray 3” option 604 in set paper source option box 316 as the paper source for the selected page range; and the media type selected box 312

indicates that the user has selected “Light Green” option 606 in set media type option box 318 as the media type for the selected page range.

Once again, after entering the specific attributes that are to be used for the current page-range in multiple page-range printing interface 600, the user may select
5 the next range button 408 to enter additional page-ranges and the printing attributes that are to be applied to each of the additional page-ranges.

FIG. 7 illustrates another example of a multiple page-range printing interface 700 that may be displayed by browser process 218 on client 204 in response to user selecting the next range button 408 in multiple page-range printing interface 600. In
10 this example, the user has selected a value of eight (“8”) in begin range entry box 306 to indicate that the current page-range is to start with page eight (“8”) of the electronic document. In one embodiment, the printing mechanism uses the new value in begin range entry box 306 in determining the last page of the previous page-ranges that have been entered. For example, by entering the value “8” in begin range
15 entry box 306 of multiple page-range printing interface 700, the printing mechanism determines that the page-range associated with multiple page-range printing interface 600 consists of pages 5-7. Additionally, the printing mechanism determines that the page-range associated with multiple page-range printing interface 600 currently consists of pages 8-8.

Further illustrated in this example, the simplex/duplex selected box 308
20 indicates that the user has selected simplex option 702 in set simplex/duplex option box 314 for the selected page range; the paper source selected box 310 indicates that the user has selected “Tray 2” option 704 in set paper source option box 316 as the paper source for the selected page range; and the media type selected box 312
25 indicates that the user has selected “75 g/m² (White)” option 706 in set media type option box 318 as the media type for the selected page range.

EDITING AND PRINTING SELECTING PAGE-RANGES

As illustrated in FIG. 7 and FIG. 8, in certain embodiments, begin range entry box 306 includes a radial option 708 that allows the user to view and edit the current page-ranges in addition to the print attributes that are currently associated with each of the page-ranges. For example, FIG. 8 illustrates another example of a multiple page-range printing interface 800 in which a current set of page-range entries 802 is displayed in response to the user selecting radial option 708. As illustrated, the user has selected multiple page-ranges 802 within pages 1-8, each of which have been assigned specific print attributes (media types, paper source, etc.), as previously described in FIGs. 3-7.

In one embodiment, in response to a user selecting a page-range within the current set of page-range entries 802, a multiple page-range printing interface is displayed to the user that includes the current settings for the selected page-range. For example, in response to a user selecting page-range entry 806, multiple page-range printing interface 600 is displayed by browser process 218 on client 204. The user may then change or edit the previous selections, including the value in begin range entry box 306.

Once the user is satisfied with their selections, they may select the OK button 320 to have the selected page-range and associated print attribute information ("print data") sent to the printing device. Upon receiving the print data, the printing device generates a hard copy version based on the print data.

For example, FIG. 9A illustrates a table 900 that includes an example of the type of information that may be collected in FIGs. 3-8. As a further example, FIG. 9B illustrates a table 950 that includes another example of the type of information that may be collected by the printing mechanism based on a different set of

selections than those described in FIGs. 3-8. In this example, the set of multiple page-ranges contains a page-range entry 902 that includes pages 1-2, in addition to the other page-range entries.

Appendix A illustrates an example of the type of print data that may be generated for a laser printer for an 8-page document with multiple page-range settings as defined in Table 900. In one embodiment, in response to selecting the OK button 320, the print data is sent from client 204 to printing device 202 to generate a hard copy of the range of pages based on the selections made in FIGs. 3-8. In another embodiment, the print data is generated at the printing device and in response to selecting the OK button 320 a message is sent from client 204 to printing device 202 requesting that a hard copy be generated based on the print data.

If the user decides that they do not want to print the selected page-ranges, the user may instead select the cancel button 322 to cancel the print request. Alternatively, if the user wants to select one or more additional page-ranges, the user may select the next range button 408 to view another multiple page-range printing interface for entering additional page-range selections and corresponding print attributes.

PROCESSING PRINT REQUESTS

FIG. 10 is a flow diagram that illustrates an example of a sequence that can be used for processing multiple page-range print requests within an electronic document. The steps of FIG. 10 will be explained with reference to the components illustrated in FIGs. 2-8.

At block 1002, a print request is received from a client. For explanation purposes it shall be assumed that a print request is received at printing device 202 from client 204. For example, by interacting with client 204 a user may cause a print request to be sent from browser process 218 to server process 214 of printing device

102. In one embodiment, in response to selecting a print option associated with document application 216, browser process 218 is automatically executed to cause a print request to be sent to the currently selected printing device 202.

At block 1003, the specific printing characteristics of the selected printing
5 device are obtained. These specific characteristics may include, among other things, available media type information and paper source (i.e., tray) that contains each of the media types (block 1004) and the set of formatting options (i.e., simplex/duplex, ink color, etc.) that are available to the client user (block 1006). In one embodiment, printing device 202 is configured to allow an operator to enter information through
10 an operational panel to register the characteristic information that is specific to printing device 202 (for example, the media type information that is illustrated in set media type option box 318 in FIG. 3).

At block 1008, interface data is transmitted from the printing device to the client to cause a multiple page-range interface window to be displayed at the client.
15 In one embodiment, the interface window includes a set of selectable options for specifying a specific page-range within the electronic document and for specifying the set of printing attributes that are to be applied to the pages within the specific page-range. For example, as depicted in FIG. 4, a multiple page-range printing interface 400 is displayed by browser process 218 on client 204 which allows a user
20 to select a specific page-range and the printing attributes that are to be applied to the pages within the specific page-range.

At block 1010, the printing device waits for a response from the requesting client. If at block 1012 a response is received that indicates that the user selected the cancel button 322, then the printing sequence is ended.

25 Alternatively, if at block 1014 a response is received that indicates that the user selected the print button 320, then at block 1016, a hard copy version of the

pages within each page-range is generated by the printing device based on the printing attributes that are associated with each page-range.

5 Lastly, if at block 1018 a response is received that indicates that the user selected the next range button 408, then the information associated with the current multiple page-range interface window is stored and control proceeds to block 1008 to display another multiple page-range interface window on client 204. Conversely, if a response is not received that meets the qualifications of either blocks 1012, 1014 or 1018, then control proceeds to block 1010 to wait for further user input.

10

HARDWARE OVERVIEW

FIG. 11 is a block diagram that illustrates a computer system 1100 upon which an embodiment of the invention may be implemented. Computer system 1100 includes a bus 1102 or other communication mechanism for communicating information, and a processor 1104 coupled with bus 1102 for processing information. Computer system 1100 also includes a main memory 1106, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 1102 for storing information and instructions to be executed by processor 1104. Main memory 1106 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 1104. Computer system 1100 further includes a read only memory (ROM) 1108 or other static storage device coupled to bus 1102 for storing static information and instructions for processor 1104. A storage device 1110, such as a magnetic disk or optical disk, is provided and coupled to bus 1102 for storing information and instructions.

25 Computer system 1100 may be coupled via bus 1102 to a display 1112, such as a cathode ray tube (CRT), for displaying information to a computer user. An

input device 1114, including alphanumeric and other keys, is coupled to bus 1102 for communicating information and command selections to processor 1104.

Another type of user input device is cursor control 1116, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 1104 and for controlling cursor movement on display 1112. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

The invention is related to the use of computer system 1100 for printing multiple page-ranges within an electronic document. According to one embodiment of the invention, the multiple page-range printing mechanism is provided by computer system 1100 in response to processor 1104 executing one or more sequences of one or more instructions contained in main memory 1106. Such instructions may be read into main memory 1106 from another computer-readable medium, such as storage device 1110. Execution of the sequences of instructions contained in main memory 1106 causes processor 1104 to perform the process steps described herein. One or more processors in a multi-processing arrangement may also be employed to execute the sequences of instructions contained in main memory 1106. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

The term “computer-readable medium” as used herein refers to any medium that participates in providing instructions to processor 1104 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example,

optical or magnetic disks, such as storage device 1110. Volatile media includes dynamic memory, such as main memory 1106. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 1102.

Transmission media can also take the form of acoustic or light waves, such as those
5 generated during radio wave and infrared data communications.

Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical
10 medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 1104 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote
15 computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 1100 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector coupled to bus 1102 can receive the data carried in the infrared signal and place the data on bus
20 1102. Bus 1102 carries the data to main memory 1106, from which processor 1104 retrieves and executes the instructions. The instructions received by main memory 1106 may optionally be stored on storage device 1110 either before or after execution by processor 1104.

Computer system 1100 also includes a communication interface 1118
25 coupled to bus 1102. Communication interface 1118 provides a two-way data communication coupling to a network link 1120 that is connected to a local network

1122. For example, communication interface 1118 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 1118 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 1118 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

Network link 1120 typically provides data communication through one or more networks to other data devices. For example, network link 1120 may provide a connection through local network 1122 to a host computer 1124 or to data equipment operated by an Internet Service Provider (ISP) 1126. ISP 1126 in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet" 1128. Local network 1122 and Internet 1128 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 1120 and through communication interface 1118, which carry the digital data to and from computer system 1100, are exemplary forms of carrier waves transporting the information.

Computer system 1100 can send messages and receive data, including program code, through the network(s), network link 1120 and communication interface 1118. In the Internet example, a server 1130 might transmit a requested code for an application program through Internet 1128, ISP 1126, local network 1122 and communication interface 1118. In accordance with the invention, one such downloaded application provides for printing multiple page-ranges as described herein.

5

10

20

In addition, certain types of printer characteristics have been provided as examples in describing the invention. However, embodiments of the invention are not limited to any particular type of printer characteristic. For example, in addition to the print attribute options described above, a user may also be presented with a set
5 of color, fonts, pagination, signature layout or style attribute options that are supported by the selected printing device. Thus, the specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

In addition, in this disclosure, including in the claims, certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels are used to
10 identify certain steps. Unless specifically stated in the disclosure, embodiments of the invention are not limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to imply, specify or require a particular order of carrying out such steps.

APPENDIX
Copyright © 2000 Ricoh Co. Ltd.

5 The following is an example of an output stream in PostScript format that may be generated for a laser printer for an 8-page document with multiple page-range settings as defined in Table 900. In this example, the action commands for the Multiple Page-range Setting features are highlighted to assist the reader's understanding of the code.

.%-12345X@PJL JOB
10 @PJL ENTER LANGUAGE = POSTSCRIPT
%!PS-Adobe-3.1
%%Title: Microsoft Word - 8pages.doc.DOC
%%Creator: ADOBEPS4.DRV Version 4.31
%%CreationDate: 03/10/2000 11:00:36
15 %%For: zmyu
%%BoundingBox: (atend)
%%Pages: (atend)
%%PageOrder: Special
%%DocumentNeededResources: (atend)
20 %%DocumentSuppliedResources: (atend)
%%DocumentSuppliedFeatures: (atend)
%%DocumentData: Clean7Bit
%%LanguageLevel: 3
%%TargetDevice: (Aficio MP01) (2017.603) 1
25 %%EndComments

%%BeginDefaults
%%PageBoundingBox: 13 13 600 780
30 %%ViewingOrientation: 1 0 0 1
%%PageFeatures:
%%+ *Resolution 600dpi
%%+ *Ritransparency 1
%%+ *Smoothing True
35 %%+ *RITonerSaver False
%%+ *InputSlot AutoSelectTray
%%+ *PageSize Letter
%%+ *PageRegion Letter
%%+ *Duplex None
40 %%+ *RICollate False
%%+ *StapleWhen None
%%+ *OutputBin Upper
%%+ *RIHalftoneType 2
45 %%EndDefaults

```

%%BeginProlog
%%BeginResource: procset AdobePS_Win_Feature_Safe 4.2 0
userdict begin/lucas 21690 def/featurebegin{countdictstack lucas[]bind def
/featurecleanup{stopped{cleartomark dup lucas eq{pop exit}if}loop
5 countdictstack exch sub dup 0 gt{{end}repeat}{pop}ifelse}bind def end
%%EndResource
%%BeginResource: procset AdobePS_FatalError 4.2 0
userdict begin/FatalErrorIf{{initgraphics findfont 1 index 0 eq{exch pop}{dup
length dict begin{1 index/FID ne{def}{pop pop}ifelse}forall/Encoding{
10 ISOLatin1Encoding}stopped{StandardEncoding}if def currentdict end
/ErrFont-Latin1 exch definefont}ifelse exch scalefont setfont counttomark 3
div cvi{moveto show}repeat showpage quit}{cleartomark}ifelse}bind def end
%%EndResource
userdict begin /PrtVMMsg {[
15 (This job requires more memory than is available in this printer.) 100 500
(Try one or more of the following, and then print again:) 100 485
(In the PostScript dialog box, click Optimize For Portability.) 115 470
(In the Device Options dialog box, make sure the Available Printer Memory is
accurate.) 115 455
20 (Reduce the number of fonts in the document.) 115 440
(Print the document in parts.) 115 425
12 /Times-Roman showpage
(%%[ PrinterError: Low Printer VM ]%%) =
true FatalErrorIf} bind def end
25 %%BeginResource: procset AdobePS_Win_ErrorHandler 4.2 0
/currentpacking where{pop/oldpack currentpacking def/setpacking where{pop
false setpacking}if}/$brkpage 64 dict def $brkpage begin/prnt{dup type
/stringtype ne{=string cvs}if dup length 6 mul/tx exch def/ty 10 def
currentpoint/toy exch def/tox exch def 1 setgray newpath tox toy 2 sub moveto
30 0 ty rlineto tx 0 rlineto 0 ty neg rlineto closepath fill tox toy moveto 0
setgray show}bind def/nl{currentpoint exch pop lmargin exch moveto 0 -10
rmoveto}def/=={/cp 0 def typeprint nl}def/typeprint{dup type exec}readonly def
/lmargin 72 def/rmargin 72 def/tprnt{dup length cp add rmargin gt{nl/cp 0 def
}if dup length cp add/cp exch def prnt}readonly def/cvsprint{=string cvs
35 tprnt( )tprnt}readonly def/integertype{cvsprint}readonly def/realttype{
cvsprint}readonly def/booleantype{cvsprint}readonly def/operatortype{(--
)tprnt =string cvs tprnt(-- )tprnt}readonly def/marktype{pop(-mark- )tprnt}
readonly def/dicttype{pop(-dictionary- )tprnt}readonly def/nulltype{pop
(-null- )tprnt}readonly def/filetype{pop(-filestream- )tprnt}readonly def
40 /savetype{pop(-savelevel- )tprnt}readonly def/fonttype{pop(-fontid- )tprnt}
readonly def/nametype{dup xcheck not{(/)tprnt}if cvsprint}readonly def
/stringtype{dup rcheck {(\\)tprnt tprnt(\\)tprnt}{pop(-string- )tprnt}
ifelse}readonly def/arraytype{dup rcheck{dup xcheck{({)tprnt{typeprint}forall
( )tprnt}{(\\)tprnt{typeprint}forall(\\)tprnt}ifelse}{pop(-array- )tprnt}
45 ifelse}readonly def/packedarraytype{dup rcheck{dup xcheck{({)tprnt{typeprint}
forall(\\)tprnt}{(\\)tprnt{typeprint}forall(\\)tprnt}ifelse}{pop

```

```

(-packedarray- )tprint}ifelse}readonly def/courier/Courier findfont 10
scalefont def end errordict/handleerror{systemdict begin $error begin $brkpage
begin newerror{/newerror false store vmstatus pop pop 0 ne{grestoreall}if
showpage initgraphics courier setfont lmargin 720 moveto errorname(VMerror)eq{
5   PrtVMMsg} {(ERROR: )print errorname print nl(OFFENDING COMMAND:
)print/command
load print $error/ostack known{nl nl(STACK:)print nl nl $error/ostack get aload
length{==}repeat}if}ifelse systemdict/showpage get exec(%%[ Error: )print
errorname =print( OffendingCommand: )print/command load =print( ]%%)= flush}
10  if end end end}dup 0 systemdict put dup 4 $brkpage put bind readonly put
/currentpacking where{pop/setpacking where{pop oldpack setpacking}if}if
%%EndResource
%%BeginResource: procset AdobePS_Win_Driver_Incr_L2 4.2 0
userdict /AdobePS_Win_Driver_Incr_L2 250 dict dup begin put
15  [
(This job can print only on a PostScript(R) Level 2 or 3 ) 100 500
(printer. Please send this file to a Level 2 or 3 printer.) 100 485
( ) 100 470
12 /Times-Roman
20  /languagelevel where {pop languagelevel} {1} ifelse 2 lt FatalErrorIf
/VM? {pop} bind def
30000 VM?
%%BeginResource: procset AdobePS_Win_Utills 4.2 0
/d/def load def/,/load load d/~ /exch , d/? /ifelse , d!/ /pop , d!/ /begin , d/^
25  /index , d/@ /dup , d/+ /translate , d/$ /roll , d/U /userdict , d/M /moveto , d/-
/r/lineto , d/& /currentdict , d;/ /gsave , d;/ /grestore , d/F /false , d/T /true ,
d/N /newpath , d/E /end , d/Ac /arc , d/An /arcn , d/A /ashow , d/D /awidthshow , d
/C /closepath , d/V /div , d/O /eofill , d/L /fill , d/I /lineto , d/-C /rcurveto ,
d/-M /rmoveto , d/+S /scale , d/Ji /setfont , d/Lc /setlinecap , d/Lj /setlinejoin
30  , d/Lw /setlinewidth , d/S /show , d/LH /showpage , d/K /stroke , d/W /widthshow ,
d/R /rotate , d/XS /xshow , d/b {bind d} bind d/bd {bind d} bind d/xd {~ d} bd/ld {, d}
bd/lw /Lw ld/lc /Lc ld/lj /Lj ld/sg /setgray ld/bn /bind ld/L2? F/languagelevel
where{! languagelevel 2 ge{! T}if}if d/L3? F/languagelevel where{!
languagelevel 3 ge{! T}if}if d/g{@ not{U/Deflfl_save save put}if U/Deflfl_bool 2
35  ^ put}b/Deflfl_El{if U/Deflfl_bool get not @{U/Deflfl_save get restore}if}b/e{
Deflfl_El !}b/self & d/reinitialize{[/TextInit/GraphInit/UtilsInit counttomark{
@ where{self eq} {F}?{cvx exec} {!}?}repeat cleartomark}b/initialize{` {
/Pscript_Win_Data where{!} {U/Pscript_Win_Data & put} ?/ADO_mxRot ~ d
/TextInitialised? F d reinitialize E} {U/Pscript_Win_Data 230 dict @ ` put
40  /ADO_mxRot ~ d/TextInitialised? F d reinitialize} ?}b/terminate{! {& self eq{
exit} {E} ?}loop E}b/suspend/terminate , d/resume{` Pscript_Win_Data `}b/snap{
transform 0.25 sub round 0.25 add ~ 0.25 sub round 0.25 add ~ itransform}b
/dsnap{dtransform round ~ round ~ idtransform}b U<04>cvn{}put/setjn{{
statusdict/jobname known{statusdict/jobname 3 -1 $ put}if}stopped cleartomark}
45  b/solid{[[]0 setdash}b/setdash{0 setdash}b/colspRefresh{}b/rp{4 2 $ M 1 ^ 0 - 0
~ - neg 0 -}b/rr{1 ^ 0 - 0 ~ - neg 0 - C}b/CTMsave{globaldict ` currentglobal

```

```

T setglobal/SavedCTM matrix currentmatrix d setglobal E}b/CTMrestore{
globaldict `currentglobal T setglobal/SavedCTM where{! SavedCTM setmatrix}if
setglobal E}b/emuxs{!! currentpoint( )@ 0 6 -1 $ put S ~ 3 ^ 3 ^ get add ~ M
1 add}b/XSE{version cvi 2015 ge{XS}{0/emuxs , 4 -1 $ cshow ! !}}b/UDF{L2?{
5 undefinefont}{!}}b/UDR{L2?{undefineresource}{! !}}b/freeVM{/Courier findfont
[40 0 0 -40 0 0]makefont Ji 2 vmreclaim}b
%%EndResource
%%BeginResource: procset AdobePS_Win_Utills_L2 4.2 0
/colspA/DeviceGray d/colspABC/DeviceRGB d/setAorABC{{colspA}{colspABC}?
10 setcolorspace}b/rf/rectfill , d/fx{1 1 dtransform @ 0 ge{1 sub 0.5}{1 add -0.5
}} 3 -1 $ @ 0 ge{1 sub 0.5}{1 add -0.5}} 3 1 $ 4 1 $ idtransform 4 -2 $
idtransform}b/BZ{4 -2 $ snap + +S fx rf}b/rs/rectstroke , d/rc/rectclip , d/sg
{@ @ setcolor}b/sco{setcolor}b/colspRefresh{colspABC setcolorspace}b/sgco{{sg
}{sco}}b/UtilsInit{currentglobal{F setglobal}if}b/resourcestatus where{!
15 /ColorRendering/ProcSet resourcestatus{! ! T}{F}}{F}? not{/ColorRendering<<
/GetHalftoneName{currenthalftone @/HalftoneName known{/HalftoneName get}{!
/none}}?}bn/GetPageDeviceName{currentpagedevice @/PageDeviceName known{
/PageDeviceName get @ null eq{!/none}if}{!/none}}?}bn/GetSubstituteCRD{!
/DefaultColorRendering/ColorRendering resourcestatus{! !/DefaultColorRendering
20 }{(DefaultColorRendering*)}{cvn exit}127 string/ColorRendering resourceforall}?
}bn>>/defineresource where{!/ProcSet defineresource !}{! !}}if/buildcrdname{
/ColorRendering/ProcSet findresource `mark GetHalftoneName @ type @/nametype
ne ~/stringtype ne and{!/none}if(.).GetPageDeviceName @ type @/nametype ne ~
/stringtype ne and{!/none}if(.).5 ^ 0 5 -1 1 {^ length add}for string 6 1 $ 5 ^
25 5 {~ 1 ^ cvs length 1 ^ length 1 ^ sub getinterval}repeat ! cvn 3 1 $ ! ! E}b
/definecolorrendering{~ buildcrdname ~/ColorRendering defineresource !}b
/findcolorrendering where{!}{/findcolorrendering{buildcrdname @/ColorRendering
resourcestatus{! ! T}{/ColorRendering/ProcSet findresource `GetSubstituteCRD
E F}}?}b}/selectcolorrendering{findcolorrendering !/ColorRendering
30 findresource setcolorrendering}b/ExecWMForm{execform}b/setpagedevice where{!
/realstpgdev/setpagedevice ld}if/SC_topddict 0 d/SC_spdict 0 d/dopgdev{
Pscript_Win_Data/setpagedevice undef SC_topddict @ length 0 gt{realstpgdev}if}
bd/stpgdev{SC_topddict @ 3 -1 ${SC_spdict 2 ^ known{SC_spdict 2 ^ get @ 3 -1 $
{put @}forall ! put @}{put @}}forall ! !}bd/ststpgdev{Pscript_Win_Data
35 /setpagedevice/stpgdev , put/SC_topddict 0 dict store/SC_spdict 3 dict `
/InputAttributes 0 dict d/Policies 0 dict d/OutputAttributes 0 dict d & E
store}d/notdefXPos 0 d/notdefYPos 0 d/SetNotDefCharString{@ findfont/Encoding
get 0 get/.notdef eq{@ 1000 selectfont<00>stringwidth/notdefYPos ~ d
/notdefXPos ~ d}{/notdefXPos 0 d/notdefYPos 0 d}? findfont/CharStrings get
40 /.notdef{! notdefXPos notdefYPos setcharwidth}put}bind d
%%EndResource
%%BeginResource: procset AdobePS_Win_Clip_Emul 4.2 0
L3? not{/clipsave/: , d/cliprestore{U `curr_cs currentcolorspace d/curr_cd
currentcolorrendering d/curr_color[currentcolor]d/curr_font currentfont d
45 /curr_lc currentlinecap d/curr_lj currentlinejoin d/curr_lw currentlinewidth d
/curr_dash[currentdash]d/curr_ml currentmiterlimit d ; curr_cs setcolorspace

```



```

curr_cd setcolorrendering curr_color{}forall setcolor curr_font Ji curr_lc Lc
curr_lj Lj curr_lw Lw curr_dash{}forall setdash curr_ml setmiterlimit E}bind d
}if/cs/clipsave , d/cr/cliprestore , d
%%EndResource
5 end
%%EndResource
%%EndProlog

%%BeginSetup
10 statusdict begin (%%[ ProductName: ) print product print ( ]%%)= flush end
/findresource where{pop mark{/CIDParams /ProcSet findresource
/SetBuildCompatible get true exch exec}stopped cleartomark}if
[ 1 0 0 1 0 0 ] false AdobePS_Win_Driver_Incr_L2 dup /initialize get exec

15 %%BeginNonPPDFeature: JobTimeout 0
featurebegin{0
<</JobTimeout 3 -1 roll>>setuserparams
}featurecleanup
%%EndNonPPDFeature
20 %%BeginNonPPDFeature: WaitTimeout 240
featurebegin{240
<</WaitTimeout 3 -1 roll>>setuserparams
}featurecleanup
%%EndNonPPDFeature
25 featurebegin{ ststpgdev
%%BeginFeature: *Resolution 600dpi

(<<) cvx exec/HWResolution [600 600] /Policies
(<<) cvx exec/HWResolution 2(>>) cvx exec (>>) cvx exec setpagedevice
30 %%EndFeature
%%BeginFeature: *RItransparency 1

%%EndFeature
35 dopgdev }featurecleanup
featurebegin{ ststpgdev
%%BeginFeature: *Smoothing True

<< /PostRenderingEnhance true >> setpagedevice
40 %%EndFeature
%%BeginFeature: *RITonerSaver False

<< /TonerSave false >> setpagedevice
%%EndFeature
45 dopgdev }featurecleanup
featurebegin{

```

```

} featurecleanup
featurebegin{
%%BeginFeature: *InputSlot AutoSelectTray
5
%%EndFeature
} featurecleanup
featurebegin{
%%BeginFeature: *PageSize Letter
10
    <</Policies <</PageSize 2>> /PageSize [612 792] /ImagingBBox null>>
    setpagedevice
    %%EndFeature
} featurecleanup
15 featurebegin{ ststpgdev
    %%BeginFeature: *Duplex None

        <</Duplex false /Tumble false>> setpagedevice
    %%EndFeature
20 %%BeginFeature: *RICollate False
    1 dict dup /Collate false put setpagedevice
    %%EndFeature
    %%BeginFeature: *StapleWhen None
    1 dict dup /Staple 0 put setpagedevice
25 %%EndFeature
    %%BeginFeature: *OutputBin Upper
    <</OutputType (Standard Tray)>> setpagedevice
    %%EndFeature
    %%BeginFeature: *RIHalftoneType 2
30
    <<
        /Install
        [
35         currentpagedevice /Install 2 copy known
        {
            get /exec cvx
        }{
            pop pop
        }ifelse
40     {
        <<
            300 /Ricoh300HT60
            400 /Ricoh400HT80
            600 /Ricoh600HT106
45         >>
            currentpagedevice /HWResolution get 0 get get

```

```

    /Halftone findresource dup length dict copy sethalftone
    }
    /exec cvx
    ]cvx bind
5  >>setpagedevice
    %%EndFeature
    dopgdev }featurecleanup

    %%% Begin Multiple Page Setting:
10  %%% Z. Yu Begin: *InputSlot Tray2/Tray 2
    gsave
    featurebegin{
    %%BeginFeature: *InputSlot Tray2
    currentpagedevice
15  /InputAttributes get dup 1 known
    {1 get dup null eq {pop}
    {dup length 1 add dict copy dup /InputAttributes
    1 dict dup
    currentpagedevice /InputAttributes get /Priority get 0 get 1 exch
20  6 0 2 3 5 7 8 array astore /Priority exch
    put
    put setpagedevice} ifelse <</TraySwitch false>> setpagedevice
    } {pop} ifelse
    %%EndFeature
25  }featurecleanup
    grestore
    %%% Z. Yu End: *InputSlot Tray2/Tray 2

    %%% Z. Yu Begin: *Duplex None
30  featurebegin{
    %%BeginFeature: *Duplex None
    <</Duplex false /Tumble false>> setpagedevice
    %%EndFeature
    }featurecleanup
35  %%% Z. Yu End: *Duplex None
    %%% End Multiple Page Setting:

    1 setlinecap 1 setlinejoin
    /mysetup [ .12 0 0 -.12 13 780 ] def
40  userdict begin /savelevel0 save def end
    %%EndSetup

    %%Page: 1 1

45  %%EndPageComments
    %%BeginPageSetup
```

```

featurebegin{ ststpgdev
%%BeginFeature: *Option7 True

%%EndFeature
5  %%BeginFeature: *Option6 True

%%EndFeature
%%BeginFeature: *Option5 True

10  %%EndFeature
%%BeginFeature: *Option4 True

%%EndFeature
%%BeginFeature: *Option3 False
15  %%EndFeature
%%BeginFeature: *Option2 True

%%EndFeature
20  %%BeginFeature: *Option1 True

%%EndFeature
%%BeginFeature: *VMOption 8Meg
25  %%EndFeature
dopgdev }featurecleanup
mysetup concat [ matrix currentmatrix
    {dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh
%%EndPageSetup
30
AdobePS_Win_Driver_Incr_L2 begin
%%BeginResource: procset AdobePS_Win_Text 4.2 0
/TextInit{TextInitialised? not{/Pscript_Windows_Font & d/TextInitialised? T d
/fM[1 0 0 -1 0 0]d/mFM matrix d/mAFM matrix d/iMat[1 0 0.212557 neg 1 0 0]d}if
35  }b/copyfont{1 ^ length add dict ` {1 ^/FID ne{d} {! !}??}forall & E}b/EncodeDict
9 dict d/bullets{ {/bullet}repeat}b/rF{3 copyfont @ ` ~ EncodeDict ~ get
/Encoding ~ &/CharStrings known{CharStrings/Eth known not{! EncodeDict
/ANSIEncodingOld get}if}if d E}b/FDV/FDepVector d/pd_scratch 128 string d
/pd_concatnames{2 copy cvs length 3 ^ 2 ^ cvs length @ 3 1 $ add 1 add string
40  @ @ @ 0 9 -1 $ 7 ^ cvs putinterval 3 ^ 45 put 5 -2 $ cvs 4 -1 $ 1 add ~
putinterval cvn}b/pd_genunqname{pd_Incr @ 1 add/pd_Incr ~ d pd_scratch cvs cvn
pd_InstName ~ pd_scratch
pd_concatnames}b/pd_GetAdoNotDefFont{U(AdoNotDefFont)
2 copy known{get} { @ 11 dict `/FontName 1 ^ d/FontMatrix matrix d/FontType 0 d
45  /FMapType 2 d/Encoding[0 0 0 0]d/FDepVector[/NotDefFont findfont]d & E
definefont @ 4 1 $ put}??}b/pd_FCIsCovered{ @/SubsVector get @ 0 ~{add}forall

```

```
256 ge{! ! T} {length 1 sub ~ /Encoding get ~ get F}??b/pd_CoverFCRange{@
pd_FCIsCovered not{~ @ FDV 2 copy get @ 6 -1 $ pd_GetAdoNotDefFont put
put}if}
b/pd_RKSJ2Ext{{{(Ext-RKSJ-H)}}{(Ext-RKSJ-V)}}? ~ @ length @ 6 sub ~ 4 add
5 string
@ @ 0 6 -1 $ @ length string cvs putinterval 4 2 $ ~ putinterval cvn}b
/pd_FindExtFont{pd_RKSJ2Ext @ {findfont}stopped{! ! F} {@/FontName 2 copy
known{
get 3 -1 $ eq @ not{~ !}if} {4{!}repeat F}??}b/pd_AddEm87{pd_FindExtFont{FDV
10 get 2 get FDV get 7 get ~ @ FDV 2 copy get @ length array copy @ 2 2 copy get
0 copyfont @ FDV 2 copy get @ length @ @ 1 add array @ 0 6 -1 $ putinterval @
4 2 $ 6 1 $ 14 -1 $ put put 1 ^ /Encoding 2 copy get @ length array copy @ 6 6
-1 $ put put @/FontName pd_genunqname @ 4 1 $ put ~ definefont put put}if
pd_CoverFCRange}b/mAF{mAFM 5 3 -1 $ put mAFM makefont
15 Pscript_Windows_Font 3 1
$ put}b/mF{@ 7 1 $ findfont ~{@ /Encoding get @ StandardEncoding eq{! T} {{
ISOLatin1Encoding}stopped{! F} {eq}??{T} {@ ` T 32 1 127 {Encoding 1 ^ get
StandardEncoding 3 -1 $ get eq and}for E}??} {F}??{1 ^ ~ rF} {0 copyfont}? 6 -2
${T pd_AddEm87 ~ !} {! ~ !/pd_charset @ where{~ get 128 eq{@ FDV 2 copy get @
20 length array copy put pd_CoverFCRange}if} {!}??? 2 ^ ~ definefont fM 5 4 -1 $
put fM 4 0 put fM makefont Pscript_Windows_Font 3 1 $ put}b
/pd_IsModeSwitchAble{/resourcestatus where{!/CIDParams/ProcSet 2 copy
resourcestatus{! ! findresource @/SetBuildCompatible known ~
/GetBuildCompatible known and ~ 0}if ! !}if}b/pd_LParams 8 dict d
25 /pd_DefLocalParams{pd_LParams `pd_InstName ~ d/pd_Incr 0 d @/pd_charset ~ d
/pd_SwitchAble pd_IsModeSwitchAble d/pd_PreviousMode T d ! & E}b
/pd_IsCID-KeyedFont{/resourcestatus where{!/CMap resourcestatus}stopped{! ! !
F} {{! ! /CIDFont resourcestatus{! ! T} {F}??} {! F}??} {! ! F}??}b
/pd_SwitchToNative{F pd_SwitchAble{!/CIDParams/ProcSet findresource @
30 /GetBuildCompatible get exec F 3 -1 $ /SetBuildCompatible get exec}if
/pd_PreviousMode ~ d}b/pd_IsCIDCapable{/CIDInit/ProcSet resourcestatus @{! ! !
T}if}b/pd_mF_Finally{fM @ 4 0 put @ 5 5 -1 $ put makefont Pscript_Windows_Font
3 1 $ put}b/pd_SwitchToPrevious{pd_SwitchAble{pd_PreviousMode/CIDParams
/ProcSet findresource/SetBuildCompatible get exec}if}b/pd_gen90ms{{
35 /90ms-RKSJ-H} {/90ms-RKSJ-V}? pd_scratch pd_concatnames}b/pd_GetHKPos{@
/SubsVector get @ length 1 sub 0 1 ^ 1 1 3 -1 $ {~ 3 ^ 2 ^ get add @ 160 eq{4 2
$ ! ! ! exit} {~ 2 ^ ge{! ! ! 0}if}??}for ~ /Encoding get ~ get}b/mF_83pv_CID{T
pd_gen90ms[4 ^ [5 -2 $ ~]/CIDInit/ProcSet findresource ` beginrearrangedfont 1
usefont 2 beginbfrange<00><80><00><a0><df><a0>endbfrange endrearrangedfont E
40 cleartomark 1 ^ findfont}b/mF_83pv_CSL{T pd_gen90ms ~ findfont @
pd_GetHKPos ~
FDV get @ 0 get 3 1 $ ~ get 3 -1 $ findfont 0 copyfont @ pd_GetHKPos ~ @ FDV 2
copy get @ length array copy @ 0 9 -1 $ 2 ^ 10 -2 $ ~ put put put 2 ^ ~
definefont}b/mF_FE{6 -1 $ ! 6 ^ pd_DefLocalParams ` 2 copy pd_IsCID-KeyedFont{
45 4 -1 $ ! 6 -1 $ {3 -1 $ ! ! /90ms-RKSJ-H 2 copy pd_scratch pd_concatnames 3 1 $}
if pd_SwitchToNative @/83pv-RKSJ-H eq{! pd_IsCIDCapable{mF_83pv_CID}{
```

```
mF_83pv_CSL}}{4 ^ ~[4 -1 $]composefont ~ !}? pd_mF_Finally
pd_SwitchToPrevious}{! !/0 3 1 $ mF}? E}b/xF{scalefont Pscript_Windows_Font 3
1 $ put}b/xMF{mFM astore makefont Pscript_Windows_Font 3 1 $ put}b/xF2
/scalefont , d/xMF2{mFM astore makefont}b/sLT{: Lw -M currentpoint snap M 0 -
5 0 Lc K ;}b/xUP null d/yUP null d/uW null d/xSP null d/ySP null d/sW null d/sSU
{N/uW ~ d/yUP ~ d/xUP ~ d}b/sU{xUP yUP uW sLT}b/sST{N/sW ~ d/ySP ~ d/xSP
~ d}b
/sT{xSP ySP sW sLT}b/sR{: + R 0 0 M}b/sRxy{: matrix astore concat 0 0 M}b/eR/;
, d/AddOrigFP{{&/FontInfo known{&/FontInfo get length 6 add}{6}? dict `
10 /WinPitchAndFamily ~ d/WinCharSet ~ d/OrigFontType ~ d/OrigFontStyle ~ d
/OrigFontName ~ d & E/FontInfo ~ d}{! ! ! ! !}}b/G00GFFEncoding[/G00/G01/G02
/G03/G04/G05/G06/G07/G08/G09/G0A/G0B/G0C/G0D/G0E/G0F/G10/G11/G12/G1
3/G14/G15
/G16/G17/G18/G19/G1A/G1B/G1C/G1D/G1E/G1F/G20/G21/G22/G23/G24/G25/G2
15 6/G27/G28
/G29/G2A/G2B/G2C/G2D/G2E/G2F/G30/G31/G32/G33/G34/G35/G36/G37/G38/G3
9/G3A/G3B
/G3C/G3D/G3E/G3F/G40/G41/G42/G43/G44/G45/G46/G47/G48/G49/G4A/G4B/G4
C/G4D/G4E
20 /G4F/G50/G51/G52/G53/G54/G55/G56/G57/G58/G59/G5A/G5B/G5C/G5D/G5E/G5
F/G60/G61
/G62/G63/G64/G65/G66/G67/G68/G69/G6A/G6B/G6C/G6D/G6E/G6F/G70/G71/G7
2/G73/G74
/G75/G76/G77/G78/G79/G7A/G7B/G7C/G7D/G7E/G7F/G80/G81/G82/G83/G84/G8
25 5/G86/G87
/G88/G89/G8A/G8B/G8C/G8D/G8E/G8F/G90/G91/G92/G93/G94/G95/G96/G97/G9
8/G99/G9A
/G9B/G9C/G9D/G9E/G9F/GA0/GA1/GA2/GA3/GA4/GA5/GA6/GA7/GA8/GA9/GA
A/GAB/GAC/GAD
30 /GAE/GAF/GB0/GB1/GB2/GB3/GB4/GB5/GB6/GB7/GB8/GB9/GBA/GBB/GBC/G
BD/GBE/GBF/GC0
/GC1/GC2/GC3/GC4/GC5/GC6/GC7/GC8/GC9/GCA/GCB/GCC/GCD/GCE/GCF/G
D0/GD1/GD2/GD3
/GD4/GD5/GD6/GD7/GD8/GD9/GDA/GDB/GDC/GDD/GDE/GDF/GE0/GE1/GE2/
35 GE3/GE4/GE5/GE6
/GE7/GE8/GE9/GEA/GEB/GEC/GED/GEE/GEF/GF0/GF1/GF2/GF3/GF4/GF5/GF6/
GF7/GF8/GF9
/GFA/GFB/GFC/GFD/GFE/GFF]readonly d
%%EndResource
40
%%BeginResource: procset AdobePS_Win_Encoding 4.2 0
/ANSIEncoding[/grave/acute/circumflex/tilde/macron/breve/dotaccent/dieresis
/ring/cedilla/hungarumlaut/ogonek/caron/dotlessi 18 bullets StandardEncoding
32 95 getinterval aload !/notdef/Euro/notdef/quotesinglbase/florin
45 /quotedblbase/ellipsis/dagger/daggerdbl/circumflex/perthousand/Scaron
/guilsinglleft/OE/notdef/Zcaron/notdef/notdef/quoteleft/quoteright
```

```

/quotedblleft/quotedblright/bullet/endash/emdash/tilde/trademark/scaron
/guilsinglright/oe/.notdef/zcaron/Ydieresis{ISOLatin1Encoding}stopped{96
bullets}{160 96 getinterval aload !}?]d ANSIEncoding @ 39/quotesingle put 96
/grave put/ANSIEncodingOld ANSIEncoding 256 array copy d ANSIEncodingOld
5 @[138
153 154 169 172 174 177 178 179 181 185 188 189 190 208 215 221 222 240 247
253 254]}{/bullet put @}forall 166/bar put 176/ring put EncodeDict/0
ANSIEncoding put EncodeDict/ANSIEncodingOld ANSIEncodingOld put
%%EndResource
10
end reinitialize

AdobePS_Win_Driver_Incr_L2 begin
%%BeginResource: procset AdobePS_Win_Euro_L2 4.2 0
15 /UseT3EuroFont{/currentdistillerparams where{pop currentdistillerparams
/CoreDistVersion get 4000 le}{false}ifelse}bind def/NewEuroT3Font?{dup
/FontType get 3 eq{dup/EuroFont known exch/BaseFont known and}{pop false}
ifelse}bind def/T1FontHasEuro{dup/CharStrings known not{dup
NewEuroT3Font?{dup
20 /EuroGlyphName get exch/EuroFont get/CharStrings get exch known}{true}{false}
ifelse}{pop false}ifelse}{dup/FontType get 1 eq{/CharStrings get/Euro known}{
dup/InfoDict known{/InfoDict get/Euro known}{/CharStrings get/Euro known}
ifelse}ifelse}ifelse}bind def/FontHasEuro{findfont dup/Blend known{pop true}{
T1FontHasEuro}ifelse}bind def/EuroEncodingIdx 1 def/EuroFontHdr{12 dict begin
25 /FontInfo 10 dict dup begin/version(001.000)readonly def/Notice
(Copyright \(\c\)) 1999 Adobe Systems Incorporated. All Rights Reserved.)
readonly def/FullName(Euro)readonly def/FamilyName(Euro)readonly def/Weight
(Regular)readonly def/isFixedPitch false def/ItalicAngle 0 def
/UnderlinePosition -100 def/UnderlineThickness 50 def end readonly def
30 /FontName/Euro def/Encoding 256 array 0 1 255{1 index exch/.notdef put}for def
/PaintType 0 def/FontType 1 def/FontMatrix[0.001 0 0 0.001 0 0]def/FontBBox{
-25 -23 1500 804}readonly def currentdict end dup/Private 20 dict dup begin/ND
{def}def/NP{put}def/lenIV -1 def/RD{string currentfile exch readhexstring pop}
def/-|{string currentfile exch readstring pop}executeonly def/|-{def}
35 executeonly def/|{put}executeonly def/BlueValues[-20 0 706 736 547 572]|-
/OtherBlues[-211 -203]|-/BlueScale 0.0312917 def/MinFeature{16 16}|-/StdHW[60]
|-/StdVW[71]|-/ForceBold false def/password 5839 def/Erode{8.5 dup 3 -1 roll
0.1 mul exch 0.5 sub mul cvi sub dup mul 71 0 dtransform dup mul exch dup mul
add le{pop pop 1 1}{pop pop 0 1.5}ifelse}def/OtherSubrs[{}]{systemdict
40 /internaldict known not{pop 3}{1183615869 systemdict/internaldict get exec dup
/startlock known{/startlock get exec}{dup/strtlck known{/strtlck get exec}{pop
3}ifelse}ifelse}ifelse}executeonly]|-/Subrs 5 array dup 0
<8e8b0c100c110c110c210b>put dup 1<8b8c0c100b>put dup 2<8b8d0c100b>put dup
3
45 <0b>put dup 4<8e8c8e0c100c110a0b>put |- 2 index/CharStrings 256 dict dup begin
/.notdef<8b8b0d0e>def end end put put dup/FontName get exch definefont pop}

```

```
bind def/AddEuroGlyph{2 index exch EuroEncodingIdx 1 eq{EuroFontHdr}if
systemdict begin/Euro findfont dup dup/Encoding get 5 1 roll/Private get begin
/CharStrings get dup 3 index known{pop pop pop pop end end}{begin 1 index exch
def end end end EuroEncodingIdx dup 1 add/EuroEncodingIdx exch def exch put}
5 ifelse}bind def/GetNewXUID{currentdict/XUID known{[7 XUID aload pop]true}{
currentdict/UniqueID known{[7 UniqueID]true}{false}ifelse}ifelse}bind def
/BuildT3EuroFont{exch 16 dict begin dup/FontName exch def findfont dup
/Encoding get/Encoding exch def dup length 1 add dict copy dup/FID undef begin
dup dup/FontName exch def/Encoding 256 array 0 1 255{1 index exch/.notdef put}
10 for def/GetNewXUID{/XUID exch def}if currentdict end definefont pop/BaseFont
exch findfont 1000 scalefont def/EuroFont exch findfont 1000 scalefont def pop
/EuroGlyphName exch def/FontType 3 def/FontMatrix[0.001 0 0 0.001 0 0]def
/FontBBox BaseFont/FontBBox get def/Char 1 string def/BuildChar{exch dup begin
/Encoding get 1 index get/Euro eq{BaseFont T1FontHasEuro{false}{true}ifelse}{
15 false}ifelse{EuroFont setfont pop userdict/Idx 0 put EuroFont/Encoding get{
EuroGlyphName eq{exit}{userdict/Idx Idx 1 add put}ifelse}forall userdict/Idx
get}{dup dup Encoding exch get BaseFont/Encoding get 3 1 roll put BaseFont
setfont}ifelse Char 0 3 -1 roll put Char stringwidth newpath 0 0 moveto Char
true charpath flattenpath pathbbox setcachedevice 0 0 moveto Char show end}
20 bind def currentdict end dup/FontName get exch definefont pop}bind def
/T1EuroProc{EuroFont setfont pop EuroGBBox aload pop setcachedevice 0 0 moveto
EuroGName glyphshow}def/AddEuroToT1Font{dup findfont dup length 10 add dict
copy dup/FID undef begin/EuroFont 3 -1 roll findfont 1000 scalefont def
CharStrings dup length 1 add dict copy begin/Euro{EuroFont setfont pop
25 EuroGBBox aload pop setcachedevice 0 0 moveto EuroGName glyphshow}bind def
currentdict end/CharStrings exch def/GetNewXUID{/XUID exch def}if 3 1 roll
/EuroGBBox exch def/EuroGName exch def currentdict end definefont pop}bind def
/UseObliqueEuro{findfont/FontMatrix get dup 2 get 0 eq exch dup 0 get exch 3
get eq and UseT3EuroFont or}bind def
30 %%EndResource

end reinitialize
/Times-Roman FontHasEuro not
{
35 5000 VM?
/Euro.Times-Roman
[500 0 24 -14 493 675 ]
<
A3F8880DC0EB03F854F743950C0C037DA501F7A0BB01F805BB01F91EA501F86
40 9
F8911584F73A0576068478827E7F8B087F8B70AC438B082A8BFB0D6471FB7008
63067C5B05BE068A7E8B878B7D08750764067D5B05C206A0FB7BF71858E88B0
8
8F0AC0EB03F84BF789950C0C037DA501F7A0BB01F805BB01F91EA501CA8BA
45 CAC
998B08988B95809076088F0AC0EB03F854F743950C0C037DA501F7A0BB01F805
```



```
BB01F91EA501A00692F73705750675FB094176678B083A8B43C182F75E08F774
0699BB05FB84069C078B978C978B9708F7950699BB05FBA1069AF755D5B3D18
B
08AB8BDD71A2FB0708090E
5  >
    AddEuroGlyph /Euro /Times-Roman
    UseT3EuroFont
    {/Times-Roman-Copy BuildT3EuroFont}
    {AddEuroToT1Font} ifelse
10  } if
    %%IncludeResource: font Times-Roman
    false /F0 .8986 /0 true (Times-Roman) cvn mF
    /F0S217 F0 217 xF
    F0S217 Ji
15  1 1 0 sco 642 505 M
    (Page 1)[ 121 96 109 96 54 109 ] XSE
    642 754 M
    (White)[ 205 109 60 60 96 ] XSE
    642 1003 M
20  (Tray 2)[ 133 72 96 109 54 109 ] XSE
    642 1253 M
    (Simplex)[ 121 60 168 109 60 96 109 ] XSE

    LH
25  (%%[ Page: 1 ]%%) = flush
    %%PageTrailer

    %%% Begin Multiple Page Setting:
30  %%% Z. Yu *InputSlot Manual/Bypass Tray
    gsave
    currentpagedevice
    /InputAttributes get dup 5 known
    {5 get dup null eq {pop}
35  {dup length 1 add dict copy dup /InputAttributes
        1 dict dup
        currentpagedevice /InputAttributes get /Priority get 0 get 5 exch
        6 0 1 2 3 7 8 array astore /Priority exch
        put
40  put setpagedevice} ifelse <</TraySwitch false>> setpagedevice
    } {pop} ifelse
    %%% Z. Yu End: *InputSlot Manual/Bypass Tray

    %%% Z. Yu Begin: *Duplex None
45  featurebegin{
    %%BeginFeature: *Duplex DuplexNoTumble
```

```

    <</Duplex true /Tumble false>> setpagedevice
%%EndFeature
}featurecleanup
grestore
5  %%% Z. Yu End: *Duplex DuplexNoTumble
   %%% End Multiple Page Setting:

%%Page: 2 2
10 %%EndPageComments
   %%BeginPageSetup
   featurebegin{ ststpgdev
   %%BeginFeature: *Option7 True
15  %%EndFeature
   %%BeginFeature: *Option6 True

   %%EndFeature
20  %%BeginFeature: *Option5 True

   %%EndFeature
   %%BeginFeature: *Option4 True

25  %%EndFeature
   %%BeginFeature: *Option3 False

   %%EndFeature
   %%BeginFeature: *Option2 True
30  %%EndFeature
   %%BeginFeature: *Option1 True

   %%EndFeature
35  %%BeginFeature: *VMOption 8Meg

   %%EndFeature
   dopgdev }featurecleanup
   mysetup concat [ matrix currentmatrix
40   {dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh
   %%EndPageSetup

F0S217 Ji
.7529 .7529 .7529 sco 642 505 M
45 (Page 2)[ 121 96 109 96 54 109 ] XSE
   642 754 M
```

(Gray)[157 72 96 109] XSE

642 1003 M

(Bypass Tray)[145 108 109 96 84 84 54 133 72 96 109] XSE

642 1253 M

5 (Duplex)[157 108 109 60 96 109] XSE

LH

10 (%%[Page: 2]%%) = flush
%%PageTrailer

%%Page: 3 3

15 %%EndPageComments
%%BeginPageSetup
featurebegin{ ststpgdev
%%BeginFeature: *Option7 True

20 %%EndFeature
%%BeginFeature: *Option6 True

%%EndFeature
%%BeginFeature: *Option5 True

25 %%EndFeature
%%BeginFeature: *Option4 True

30 %%EndFeature
%%BeginFeature: *Option3 False

%%EndFeature
%%BeginFeature: *Option2 True

35 %%EndFeature
%%BeginFeature: *Option1 True

%%EndFeature
%%BeginFeature: *VMOption 8Meg

40 %%EndFeature
dopgdev }featurecleanup
mysetup concat [matrix currentmatrix
{dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh
%%EndPageSetup

45 F0S217 Ji

.7529 .7529 .7529 sco 642 505 M
(Page 3)[121 96 109 96 54 109] XSE
642 754 M
(Gray)[157 72 96 109] XSE
5 642 1003 M
(Bypass Tray)[145 108 109 96 84 84 54 133 72 96 109] XSE
642 1253 M
(Duplex)[157 108 109 60 96 109] XSE

10 LH

(%%[Page: 3]%%) = flush
%%PageTrailer

15 %%Page: 4 4

%%EndPageComments
%%BeginPageSetup
featurebegin{ ststpgdev
20 %%BeginFeature: *Option7 True

%%EndFeature
%%BeginFeature: *Option6 True

25 %%EndFeature
%%BeginFeature: *Option5 True

%%EndFeature
%%BeginFeature: *Option4 True
30

%%EndFeature
%%BeginFeature: *Option3 False

%%EndFeature
35 %%BeginFeature: *Option2 True

%%EndFeature
%%BeginFeature: *Option1 True

40 %%EndFeature
%%BeginFeature: *VMOption 8Meg

%%EndFeature
dopgdev }featurecleanup
45 mysetup concat [matrix currentmatrix
{dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh

%%EndPageSetup

F0S217 Ji

.7529 .7529 .7529 sco 642 505 M

5 (Page 4)[121 96 109 96 54 109] XSE

642 754 M

(Gray)[157 72 96 109] XSE

642 1003 M

(Bypass Tray)[145 108 109 96 84 84 54 133 72 96 109] XSE

10 642 1253 M

(Duplex)[157 108 109 60 96 109] XSE

LH

15 (%%[Page: 4]%%) = flush

%%PageTrailer

%%% Begin Multiple Page Setting:

%%% page feed for the last odd page number

20

copypage

%%% Z. Yu *InputSlot Tray4/Tray 4

gsave

25 currentpagedevice

/InputAttributes get dup 3 known

{3 get dup null eq {pop}

{dup length 1 add dict copy dup /InputAttributes

1 dict dup

30 currentpagedevice /InputAttributes get /Priority get 0 get 3 exch

6 0 1 2 5 7 8 array astore /Priority exch

put

put setpagedevice} ifelse <</TraySwitch false>> setpagedevice

{pop} ifelse

35 %%% Z. Yu End: *InputSlot Tray2/Tray 2

%%% Z. Yu Begin: *Duplex None

featurebegin{

%%BeginFeature: *Duplex None

40 <</Duplex false /Tumble false>> setpagedevice

%%EndFeature

}featurecleanup

grestore

%%% Z. Yu End: *Duplex None

45 %%% End Multiple Page Setting:

%%Page: 5 5

%%EndPageComments

%%BeginPageSetup

5 featurebegin{ ststpgdev

%%BeginFeature: *Option7 True

%%EndFeature

10 %%BeginFeature: *Option6 True

%%EndFeature

%%BeginFeature: *Option5 True

%%EndFeature

15 %%BeginFeature: *Option4 True

%%EndFeature

%%BeginFeature: *Option3 False

20 %%EndFeature

%%BeginFeature: *Option2 True

%%EndFeature

%%BeginFeature: *Option1 True

25 %%EndFeature

%%BeginFeature: *VMOption 8Meg

%%EndFeature

30 dopgdev } featurecleanup

mysetup concat [matrix currentmatrix

{dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh

%%EndPageSetup

35 F0S217 Ji

0 .502 0 sco 642 505 M

(Page 5)[121 96 109 96 54 109] XSE

642 754 M

(Green)[157 72 96 96 109] XSE

40 642 1003 M

(Tray 4)[133 72 96 109 54 109] XSE

642 1253 M

(Simplex)[121 60 168 109 60 96 109] XSE

45 LH

```

(%%[ Page: 5 ]%%) = flush
%%PageTrailer

%%Page: 6 6
5  %%EndPageComments
   %%BeginPageSetup
   featurebegin{ ststpgdev
10  %%BeginFeature: *Option7 True
   %%EndFeature
   %%BeginFeature: *Option6 True

   %%EndFeature
15  %%BeginFeature: *Option5 True

   %%EndFeature
   %%BeginFeature: *Option4 True

   %%EndFeature
20  %%BeginFeature: *Option3 False

   %%EndFeature
   %%BeginFeature: *Option2 True

   %%EndFeature
25  %%BeginFeature: *Option1 True

   %%EndFeature
30  %%BeginFeature: *VMOption 8Meg

   %%EndFeature
   dopgdev }featurecleanup
   mysetup concat [ matrix currentmatrix
35   {dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh
   %%EndPageSetup

F0S217 Ji
0 .502 0 sco 642 505 M
40 (Page 6)[ 121 96 109 96 54 109 ] XSE
   642 754 M
   (Green)[ 157 72 96 96 109 ] XSE
   642 1003 M
   (Tray 4)[ 133 72 96 109 54 109 ] XSE
45 642 1253 M
   (Simplex)[ 121 60 168 109 60 96 109 ] XSE

```

LH

```
5  (%[% Page: 6 ]%) = flush
   %%PageTrailer

   %%Page: 7 7

   %%EndPageComments
10  %%BeginPageSetup
   featurebegin{ stpgdev
   %%BeginFeature: *Option7 True

   %%EndFeature
15  %%BeginFeature: *Option6 True

   %%EndFeature
   %%BeginFeature: *Option5 True

   %%EndFeature
20  %%BeginFeature: *Option4 True

   %%EndFeature
   %%BeginFeature: *Option3 False
25  %%EndFeature

   %%BeginFeature: *Option2 True

   %%EndFeature
30  %%BeginFeature: *Option1 True

   %%EndFeature
   %%BeginFeature: *VMOption 8Meg

   %%EndFeature
35  %%EndFeature
   dopgdev } featurecleanup
   mysetup concat [ matrix currentmatrix
   {dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh
   %%EndPageSetup
40  F0S217 Ji
   0 .502 0 sco 642 505 M
   (Page 7)[ 121 96 109 96 54 109 ] XSE
   642 754 M
45  (Green)[ 157 72 96 96 109 ] XSE
   642 1003 M
```


(Tray 4)[133 72 96 109 54 109] XSE
642 1253 M
(Simplex)[121 60 168 109 60 96 109] XSE

5 LH

(%%[Page: 7]%%) = flush
%%PageTrailer

10 %%% Begin Multiple Page Setting:
%%% Z. Yu Begin: *InputSlot Tray2/Tray 2
gsave
featurebegin{
%%BeginFeature: *InputSlot Tray2
15 currentpagedevice
/InputAttributes get dup 1 known
{1 get dup null eq {pop}
{dup length 1 add dict copy dup /InputAttributes
1 dict dup
20 currentpagedevice /InputAttributes get /Priority get 0 get 1 exch
6 0 2 3 5 7 8 array astore /Priority exch
put
put setpagedevice} ifelse <</TraySwitch false>> setpagedevice
} {pop} ifelse
25 %%%EndFeature
}featurecleanup
%%% Z. Yu End: *InputSlot Tray2/Tray 2

%%% Z. Yu Begin: *Duplex None
30 featurebegin{
%%BeginFeature: *Duplex None
<</Duplex false /Tumble false>> setpagedevice
%%EndFeature
}featurecleanup
35 grestore
%%% Z. Yu End: *Duplex None
%%% End Multiple Page Setting:

%%Page: 8 8
40 %%EndPageComments
%%BeginPageSetup
featurebegin{ ststpgdev
%%BeginFeature: *Option7 True
45 %%EndFeature

%%BeginFeature: *Option6 True

%%EndFeature

%%BeginFeature: *Option5 True

5 %%EndFeature

%%BeginFeature: *Option4 True

%%EndFeature

10 %%BeginFeature: *Option3 False

%%EndFeature

%%BeginFeature: *Option2 True

15 %%EndFeature

%%BeginFeature: *Option1 True

%%EndFeature

20 %%BeginFeature: *VMOption 8Meg

%%EndFeature

dopgdev } featurecleanup

mysetup concat [matrix currentmatrix

{dup dup round sub abs .0001 lt{round} if} forall] setmatrix colspRefresh

25 %%EndPageSetup

F0S217 Ji

1 1 0 sco 642 505 M

(Page 8)[121 96 109 96 54 109] XSE

30 642 754 M

(White)[205 109 60 60 96] XSE

642 1003 M

(Tray 2)[133 72 96 109 54 109] XSE

642 1253 M

35 (Simplex)[121 60 168 109 60 96 109] XSE

LH

(%%[Page: 8]%%) = flush

40 %%PageTrailer

%%Trailer

%%BoundingBox: 13 13 600 780

45 %%DocumentNeededResources:

%%+ font Times-Roman

```
%%DocumentSuppliedResources:
%%+ procset AdobePS_Win_Feature_Safe 4.2 0
%%+ procset AdobePS_FatalError 4.2 0
%%+ procset AdobePS_Win_ErrorHandler 4.2 0
5  %%+ procset AdobePS_Win_Driver_Incr_L2 4.2 0
%%+ procset AdobePS_Win_Utils 4.2 0
%%+ procset AdobePS_Win_Utils_L2 4.2 0
%%+ procset AdobePS_Win_Clip_Emul 4.2 0
%%+ procset AdobePS_Win_Text 4.2 0
10 %%+ procset AdobePS_Win_Encoding 4.2 0
%%+ procset AdobePS_Win_Euro_L2 4.2 0
%%DocumentSuppliedFeatures:
%%+ *Resolution 600dpi
%%+ *RItransparency 1
15 %%+ *Smoothing True
%%+ *RITonerSaver False
%%+ *InputSlot AutoSelectTray
%%+ *PageSize Letter
%%+ *PageRegion Letter
20 %%+ *Duplex None
%%+ *RICollate False
%%+ *StapleWhen None
%%+ *OutputBin Upper
%%+ *RIHalftoneType 2
25 %%+ JobTimeout 0
%%+ WaitTimeout 240
%%+ *Option7 True
%%+ *Option6 True
%%+ *Option5 True
30 %%+ *Option4 True
%%+ *Option3 False
%%+ *Option2 True
%%+ *Option1 True
%%+ *VMOption 8Meg
35 AdobePS_Win_Driver_Incr_L2 dup /terminate get exec
savelevel0 restore
%%Pages: 8
(%%[ LastPage ]%%) = flush
%%TrailerLength: 2169
40 %%EOF
-.-12345X@PJL EOJ
.-12345X
```